Title

Tri Nguyen

nguyetr9@oregonstate.edu

November 24, 2022

The right thing to do is Dynamic Programming. But right thing is always hard to do. So the less right things to do are

- Approximation in value space $J^*(x)$.

  For this, we have several options.

  - First, to deal with expectation is a stochastic setting, we can just assume a deterministic setting instead, by eliminating all randomness by their mode 1 value for example.

    Then in deterministic setting, for the min operator, we can use brute force, integer programming (discrete), A^* search (discrete), or nonlinear programming (continuous)

- Approximation in policy space

This is weird. In 2.1.4, they learn an approximate $\widetilde{Q}()$ based on approximate $\widetilde{J}$. I thought getting Q from J should be obvious. It may not because if action space is large.

Parameterization can be used in both value approximation and policy approximation. In both case, the setting is very similar to supervised learning setting.

RL methods has many intertwined components.

2.1.6 shows an interesting point. Good an approximation $\widetilde{J}$ is not necessarily closed to J^*. It is good as long as it is *uniformly distant* from J^*. The proposed criterion is that $Q(x, u) - \widetilde{Q}(x, u)$ change gradually as u changes.

Until now, we only talk about deterministic policy.

Good point: in the use of ℓ-lookahead, as ℓ is getting longer, the role of $\widetilde{J}$ diminishes.

A an ℓ-lookahead, we are solving ℓ-stage DP in an exact manner where we assume the terminal cost is approximated by $\widetilde{J}$.

2.3. Now we talk. Problem approximation. This is what is used in MARL currently.

Kindly think about MARL setting where environment is deterministic, policy is deterministic. could be the use of tensor?

So what so-called *policy improvement* is*rollout algorithm*. It only works under either of the following assumptions:

- The base policy is sequentially consistent, i.e., if it generates sequence of states $s_k, s_{k+1}, \dots, s_N$ starting from x_k , then it also generates a sequence $s_{k+1}, \dots, s_N$ starting from s_{k+1}.
- The base policy is sequentially improving.

Definition of rollout algorithm: It applies at state x_k the control $\widetilde{u}_k(x_k)$ given by the minimization

$$\widetilde{u}_k(x_k) \triangleq \arg\min_u \ \widetilde{Q}_k(x_k, u),$$

where $\widetilde{Q}_k(x_k, u)$ is the approximation of the true $Q_k(x_k, u)$ defined by

$$\widetilde{Q}_k(x_k, u) \triangleq g_k(x_k, u) + H_{k+1}(f_k(x_k, u)),$$

where $H_{k+1}(x_{k+1})$ is cost of the base heuristic starting from x_{k+1}.